Sensor-Assisted Global Motion Estimation for Efficient UAV Video Coding

Yang Mi†, Chunbo Luo†, Geyong Min†, Wang Miao†, Liang Wu‡, and Tianxiao Zhao+
† College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK
‡ Research Department of Multimedia, Huawei, China
+ School of Information and Electronics, Beijing Institute of Technology, China
E-mail: {ym310, c.luo, g.min, wang.miao}@exeter.ac.uk, liang.wu@huawei.com, chirs90@126.com

***Abstract*—In this paper, we propose a novel video coding scheme to significantly reduce the coding complexity and enhance overall coding efficiency in videos acquired by high mobility devices such as unmanned aerial vehicles (UAVs). In order to reduce the encoded data bits and encoding time to facilitate real-time data transmission, as well as minimize the image distortion caused by the jitter of onboard camera, a sensor-assisted global motion estimation (GMV) algorithm is designed to calculate perspective transformation model and global motion vectors, which are used in both the inter-frame coding to improve the coding efficiency and intra-frame coding to reduce block search complexity. We conducted comprehensive simulation experiments on official HM-16.10 codec and the performance results show the proposed method can achieve faster block search by 50% to 60% speedup and lower bitrate by 15% to 30% compared with standard HEVC coding software.**

***Index Terms*—HEVC, sensor-assisted global motion estimation, low delay**

I. Introduction

In recent years, one of the predominate focuses of wireless communication systems is the support of multimedia services. The use of UAVs equipped with high definition cameras is rapidly growing in the tasks such as video reconnaissance, exploitation, and surveillance. In the near future, video stream will account for over 80% of consumer Internet traffic according to [1]. The using of ultra high definition videos with high framerate and multi view will raise new challenges to video transmission tasks. Therefore, it is vital to investigate efficient video compression algorithms for wireless multimedia services.

H.265/HEVC is the latest video coding technology whose predecessor is H.264/AVC. In benchmark video compression standards, such as H.265/HEVC, global motion estimation (GME) is not adopted due to its suboptimal rate-distortion performance and complexity. The motion of each coding unit in UAV video stream is composed of camera and foreground object motion. GME is a technique that attempts to find the perceptive projection matrix between two images for video processing of high-mobility systems [2]. Differing from region ME which attempts to find the corresponding position of each individual pixel in its reference pictures, GME identifies the background motion introduced by the camera to obtain a stable and smooth video. GME requires much cheaper computation compared with local ME (LME), but can achieve high resolution image compression and transmission [3].

Image-based GME has already been integrated in MPEG-4 verification model [4] which supports translation motion. The authors of [5] suggested that the motion information can be used to merge blocks. A sensors-aided video encoding method (SaVE) that calculates the rotation movement of the camera for H.264 is proposed [6], and this method outperforms standard H.264 by 27% speedup. An approach called Sensor-assisted Motion Estimation (SaME) was proposed to estimate the global linear displacement [7], SaME focus on linear displacement estimation. A low-complexity video encoder using affine model and a matched decoder is proposed to get rid of block-level motion estimation within group of pictures (GOP) [3]. This method has proved to be outstanding in reducing bitrates but sacrifices video quality.

For many UAV videos, large translation and rotation and scaling exist between adjacent frames. The image frames might move out of the search window or image distortion caused by rotation/zoom might result in prediction unit block matching algorithms failure. The compression ratio deteriorates significantly if the block matching algorithm fails.To provide a global motion model that fits a wide range of mid-altitude UAVs, the authors of [2] propose to derived image coordinate system transform model from metadata. A low delay and low complexity video encoding for UAV inspection application is presented in [8] which replace inter-frames using homography matrix. Authors of [9], [10] also use metadata to build georeferencing model for mid-altitude fixed-wing UAVs image geometric correlation.

The major contribution of this paper includes: First, a novel method for fast motion vector prediction is proposed, and the sensor-assisted GME method is implemented based on HM-16.10 software. Second, we confirm that by performing perspective transformation, the multiscale structural similarity (MS-SSIM) between adjacent frames is increased. And last, we built our UAV video dataset with corresponding sensor log information which dataset provides an important source for future sensor assisted video codec research.

The remaining of the paper is as follows: Section II introduces the sensor-assisted video coding software framework and our method, the using of perspective transformation model and global motion vectors. Section III provides experiment results to study the performance of our framework

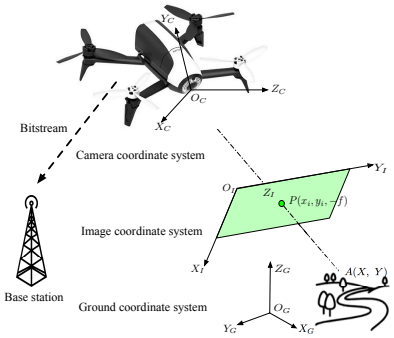

Fig. 1: The wireless communication scenario of our sensor-assisted video coding method and pinhole model.

Name	Description
X speed v_x	Horizontal speed (X-axis), in yard
Y speed v_y	Horizontal speed (Y-axis), in yard
Z speed v_z	Vertical speed, in yard
Vehicle roll θ	Euler angle (X-axis), in radians
Vehicle pitch ϕ	Euler angle (Y-axis), in radians
Vehicle heading ψ	Euler angle (Z-axis)n, in radians
Altitude d	Terrain height, in centimeters

TABLE I: Sensor parameters used in the algorithm.

under different QPs and compares the overall performance with standard HM-16.10. Finally, Section IV concludes this paper.

II. Methodology

A. System model

The deployment scenario is illustrated in Fig. 1 which shows a small UAV with onboard camera and sensors flying around a base station within transmission range. Video streams are captured by the onboard camera, while UAV and camera motions are recorded by its sensors. The raw video stream is encoded before being transmitted to the receiver via wireless channel.

Fig. 2 describes the flow diagram of the proposed methodology. The encoder takes video stream and corresponding sensor log as inputs. The perspective transformation model (homography matrix) between image coordinate system and ground coordinate system is computed and updated once new frame and sensor data arrive. Raw image frames firstly undergo perspective transformation to remove undesired distortions caused by external motion. The GME method completely relies on sensor information provided by UAV system. An frame motion monitor is used in order to determine whether large-scale motion exists between two adjacent frames, if not, the encoding process could be skipped by transmitting a 9-element homography matrix (HM) instead. Otherwise, block matching process with fast motion vector predictor (MVP) candidate list is executed to locate the best local MV. Table. I shows the sensor data used in our method.

B. Perspective transformation for UAV videos

The global motion of UAV video can be modelled by a combination of motion of the UAV and camera. To eliminate the distortion caused by those movements, a projection model is necessary for representing the relationship of image coordinate system and ground coordinate system.

The HM is modelled with intrinsic and extrinsic parameters. Intrinsic parameters describe the mapping between camera coordinates and image coordinates in the image frame, while extrinsic parameters define the location and orientation of the camera coordinate with respect to the ground coordinate (See Fig. 1).

$$K = \begin{bmatrix} \frac{1}{s_x} & 0 & c_x \\ 0 & \frac{1}{s_y} & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where (s_x, s_y) is the width and height of a single pixel, (c_x, c_y) is the horizontal and vertical offsets of camera.

Translation of the UAV between two time instances can be obtained by constructing an extrinsic matrix. We assume that the camera rotation and translation of $F(t+1)$, with respect to $F(t)$, is denotes as R and T, respectively. The rotation matrix between image coordinate system and ground coordinate system can thus be obtained as

$$R = R_\psi R_\phi R_\theta \tag{2}$$

where R_ψ, R_ϕ, and R_θ denote heading rotation, pitch rotation and roll rotation, respectively.

The displacement between image coordinate system and ground coordinate system is represented as

$$T = [t_x \ t_y \ t_z]' \tag{3}$$

Hence, the perspective transformation at time t is defined as follow:

$$H_t = K^{-1}(R + \frac{1}{d}T)K \tag{4}$$

where d denotes the altitude.

Hence, the calibrated image pair is expressed as follows

$$\begin{aligned} F_t^* &= H_t F_t \\ F_{t+1}^* &= H_{t+1} F_{t+1} \end{aligned} \tag{5}$$

C. Fast Motion Vector Predictor Estimation

After coordination transformation, the global displacement of the UAV simply obeys a linear equation $D = [dx \ dy]'$, where d_x and d_y denote the GMV at time t.

$$D^* = [dx^* \ dy^*]' = [\frac{dx}{s_x} \ \frac{dy}{s_y}]' \tag{6}$$

The translation between frame $F(t)$ and frame $F(t+1)$ can be expressed as follows

$$F_{t+1} = H_{t+1}^{-1} D H_t F_t = M F_t \tag{7}$$

In our software diagram, an image monitor based on the idea of SSIM is placed before the prediction module as shown in Fig. 3. Assuming $Ftmp = MF_t$, the SSIM between $Ftmp$ and F_{t+1} is calculated. If SSIM satisfies a given threshold, then F_{t+1} will not be processed by the encoder, but 9-element H_{t+1} will be transmitted instead.

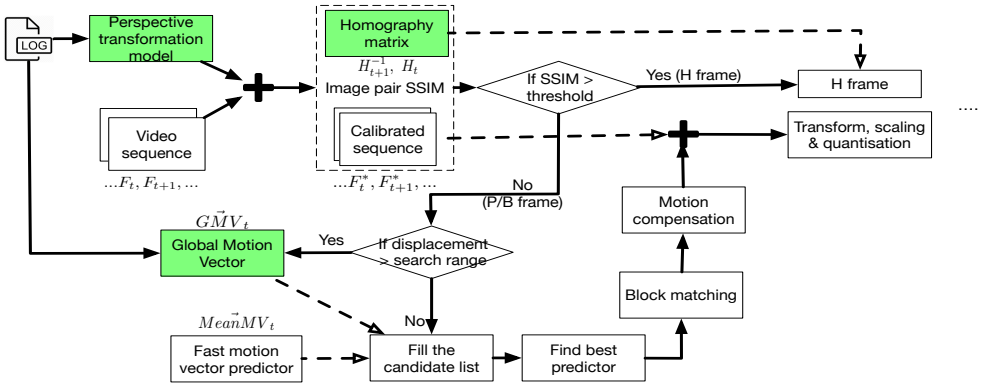

Fig. 2: The flow diagram of the proposed sensor-assisted video coding approach.

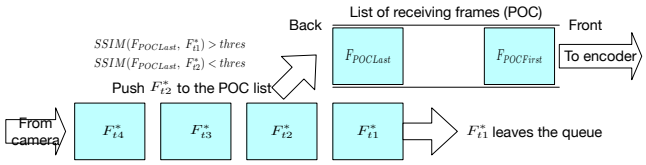

Fig. 3: The dropping out process. If two adjacent frames are considered to be similar enough, the new frame will not be processed by the encoder.

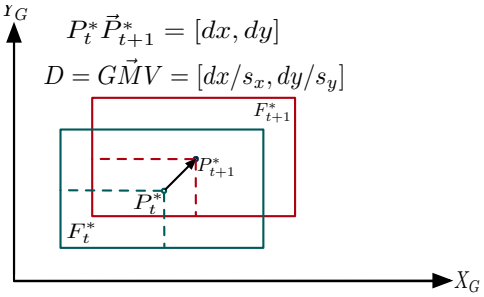

Fig. 4: The method of GMV calculation for the current block, where P^* denotes the central point of a frame.

The frame F_{t+1} can be reconstructed from H_t, F_t and H_{t+1} at the decoder end.

If the sensor information as well as the multiplication are absolutely accurate, M can represent the global motion from F_t to F_{t+1}, and thus F_t^* is similar to F_{t+1}. However, due to the UAV sensor measurement error and local motion existing between block range, in fact in most of the cases M cannot represent the best motion for each block. Therefore, the encoder has to execute block matching algorithms to find the best candidate MVs for those blocks.

We notice that in our aerial video coding process, coding blocks belonging to the background area tend to share MVs pointing to a same directory. This means that those blocks follows the global motion and the best MV of a current block has strong correlation to its spatial neighbours. Also, the blocks belonging to the foreground objects are likely to have similar LMVs of their decoded neighbours. Inspired by the work [11], we implement our fast ME process within two steps, initial candidate list and block matching search. Similar to the candidate list used in standard HEVC, our

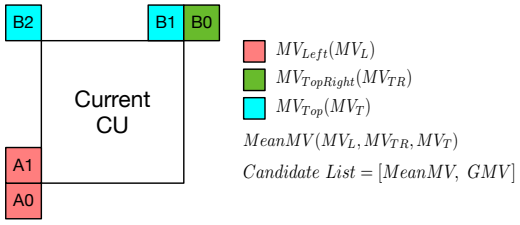

Fig. 5: The example of choosing spatial MVs from neighbouring blocks and filling the candidate list for the current block.

candidate list is filled up by two MVPs, one is the GMV from sensor information which represents the temporal correlation between reference frame and current frame (See Fig. 4), the other is a mean of neighbouring MVs (as shown in Fig. 5).

$$MeanMV = Mean(\vec{M}V_L, \vec{M}V_{TR}, \vec{M}V_T) \quad (8)$$

If the total elements in candidate list is less than two (for instance, spatial MVs are not available), a zero MV$(0,0)$ will be added to fill the candidate list. Thus, the candidate MV list (if both temporal MV and spatial MV are available) is shown as

$$Candidate\_List = \begin{cases} \vec{M}V_0 = D, & \text{GMV} \\ \vec{M}V_1 = MeanMV, & \text{LMV} \end{cases} \quad (9)$$

Fig. 5 shows the neighbouring blocks (on the left, up, and up-right) of the current block and the MVs are denotes as $\vec{M}V_L$, $\vec{M}V_{TR}$, and $\vec{M}V_T$, respectively. The best predictor is determined by Lagrangian RD cost function, block matching algorithm is performed on the TZSearch window with search range bw after the best MVP is known.

III. Experiments

The experiments were conducted on a Linux server based on the standard HEVC codec software HM-16.10. Our dataset was divided into four groups, which are $(0, 3m)$, $(3m, 6m)$, $(6m, 15m)$, and $(15m, 30m)$. The experimental results in low-altitude UAV video indicate that our method are still effective for processing videos collected at higher altitudes.

Altitude	QP	frame size (960 × 576) YPSNR	YSSIM	Bytes	Time(sec)
(0, 3m)	22	38.5230	0.9965	30679	21.5029
		37.8756	0.9963	29225	10.7863
	37	28.5315	0.9495	1927	9.2817
		28.0909	0.9428	1910	4.6961
(3m, 6m)	22	37.6262	0.9971	47709	19.7987
		37.4478	0.9951	26344	8.2494
	37	26.5360	0.9488	3473	7.2409
		26.9727	0.9315	2057	3.6333
(6m, 15m)	22	38.0860	0.9954	27942	19.6222
		37.7808	0.9939	18429	6.164
	37	27.9549	0.9339	1041	8.1853
		28.2629	0.9357	928	2.9248
(15m, 30m)	22	38.9303	0.9952	24054	20.6611
		39.4416	0.9956	16432	6.9023
	37	28.3769	0.9313	1554	8.4918
		30.1734	0.9440	961	2.5265
overall	22	38.7784	0.9954	25288	20.6549
		38.3579	**0.9951**	**19790**	**7.2743**
	37	28.8427	0.9345	1325	8.5330
		28.7104	**0.9326**	**1226**	**2.7378**

Altitude	QP	frame size (1920 × 1080) YPSNR	YSSIM	Bytes	Time(sec)
(0, 3m)	22	40.4066	0.9964	121311	107.1858
		39.2895	0.9959	100026	88.1690
	27	30.1515	0.9535	5556	102.9819
		29.6826	0.9355	5135	37.3786
(3m, 6m)	22	39.5452	0.9972	150474	92.4641
		38.8273	0.9955	146783	69.3481
	37	28.2041	0.9512	10445	34.4790
		28.7090	0.9413	6130	25.8592
(6m, 15m)	22	39.9973	0.9957	126991	87.8458
		39.1881	0.9945	92668	54.8097
	37	29.5621	0.9420	4066	34.3467
		29.3384	0.9250	4479	21.4300
(15m, 30m)	22	38.2535	0.9955	115047	87.5826
		39.3062	0.9957	98076	46.0662
	37	30.0811	0.9409	4373	43.1807
		29.6293	0.93142	4905	19.6568
overall	22	39.0130	0.9957	120602	91.6711
		39.1977	**0.9953**	**110838**	**58.0617**
	37	29.8948	0.9429	4594	33.9552
		29.3981	**0.9310**	**5571**	**20.4627**

TABLE II: The Video coding results. Results of HM-16.10 are given in the first row (in black) of each group; our results are given in the second row (in blue). YPSNR and YSSIM are used to measure the reconstructed video quality.

MaxCUHeight	64
MaxCUWidth	64
MaxCUDepth	4
Quantization Parameter (QP)	[22, 27, 32, 37]
Motion Estimation Method	TZSearch
Search Range	[-16, 16]
Coding Profile	low_delay_main_P

TABLE III: Experiment configuration conditions.

The configuration setting can be found in Table. III. To evaluate the performance of our method, we compare the quality achieved by our proposed method (in blue) and HEVC in terms of PSNR, SSIM, average bytes, and average encoding time across all P frames.

If one or more video frames is ignored by the encoder, the difference between reference frame and current frame accumulates as it can be expected. Table.II demonstrates the summary of encoding results with H frames. It can be seen that the proposed method efficiently improve the compression ratio while the image quality degradation is small. For instance, with $QP = 22$ and altitude $(6m, 15m)$, the average bytes of our sensor-assisted method is 73% of HM-16.10's, while the PSNR of Y-channel is $0.5dB$ less than HM-16.10's. The motion of camera contributes to the coding efficiency, with altitude $(3m, 6m)$ where the UAV is climbing up swiftly, the coding blocks are divided into more quad-tree blocks in HM-16.10, while the sensor-assisted HM takes advantage of the GMVs in finding matching blocks. With regard to time consumption, our algorithm can save up to 60% of the inter-frame encoding time of HM-16.10's. The second experimental result in Table.II shows that our proposed method is able to cover variety UAV movements at different definitions.

The second experiment compares the encoding performance of HM-16.10 and our method using videos of different framerate and definition (with $QP = 22$). The results shown in Fig. 6 suggest that our method supports videos under difference resolution and framerate. In the case of 1920×1080 videos at $30fps$. great benefit of

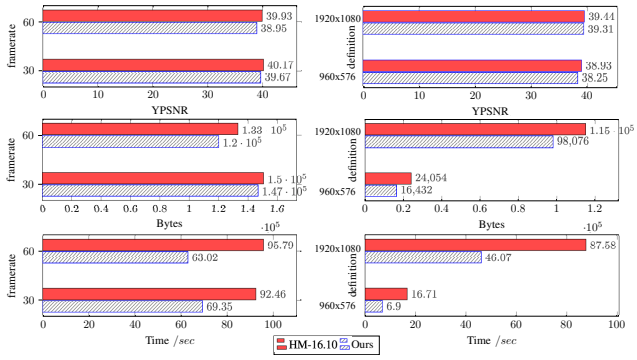

Fig. 6: Left: 1920×1080 video at $30fps$ and $60fps$, where the red bar denotes HM-16.10 and the blue pattern bar denotes ours. Right: 960×576 and 1920×1080 videos at $30fps$, where the red bar denotes HM-16.10 and the blue pattern bar denotes ours.

our method can be observed from the plot, showing 50% time saving and 15% bytes saving, at the cost of $-0.5dB$ YPSNR gain, while compared with HM-16.10's.

IV. Conclusion

In this paper, we proposed a sensor-assisted global motion estimation method aiming to trade-off reconstructed image quality and encoding time and complexity. Our experiment results show that the proposed method is able to reduce the average encoding time by 50% to 60% and the average number of bytes by 15% to 30%, while the average YPSNR and average YSSIM remain the same level, comparing with the HM-16.10. The proposed method is especially useful for low-delay or nearly real-time video transmission tasks, as well as low power platforms such as small UAVs and wireless sensor nodes in which light computation is essential.

References

[1] A. Asensio, M. Ruiz, and L. Velasco, "Requirements to support cloud, video and 5g services on the telecom cloud," in *Networks and*

Optical Communications (NOC), 2016 21st European Conference on. IEEE, 2016, pp. 64–69.

[2] H. Li, X. Li, W. Ding, and Y. Huang, "Metadata-assisted global motion estimation for medium-altitude unmanned aerial vehicle video applications," *Remote Sensing*, vol. 7, no. 10, pp. 12 606–12 634, 2015.

[3] M. Bhaskaranand and J. D. Gibson, "Low-complexity video encoding for uav reconnaissance and surveillance," in *Military Communications Conference, 2011-MILCOM 2011*. IEEE, 2011, pp. 1633–1638.

[4] S. Fukunaga, Y. Nakaya, S. H. Son, and T. Nagumo, "Mpeg-4 video verification model version 16.0," *International Organization for Standardization: Coding of Moving Pictures and Audio*, vol. 3312, pp. 1–380, 2000.

[5] H. Li, K. Fan, R. Wang, G. Li, and W. Wang, "A motion aided merge mode for hevc," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1773–1776.

[6] X. Chen, Z. Zhao, A. Rahmati, Y. Wang, and L. Zhong, "Sensor-assisted video encoding for mobile devices in real-world environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 3, pp. 335–349, 2011.

[7] L. Areekath and K. K. Palavalasa, "Sensor assisted motion estimation," in *Engineering and Systems (SCES), 2013 Students Conference on*. IEEE, 2013, pp. 1–6.

[8] R. Zhang, K. Hossain *et al.*, "Low complexity video encoding for uav inspection," in *Picture Coding Symposium (PCS), 2016*. IEEE, 2016, pp. 1–5.

[9] H. Li, X. Li, W. Ding, Y. Shi, and Y. Li, "Multi-sensor based high-precision direct georeferencing of medium-altitude unmanned aerial vehicle images," *International journal of remote sensing*, vol. 38, no. 8-10, pp. 2577–2602, 2017.

[10] H. Li, W. Ding, X. Cao, and C. Liu, "Image registration and fusion of visible and infrared integrated camera for medium-altitude unmanned aerial vehicle remote sensing," *Remote Sensing*, vol. 9, no. 5, p. 441, 2017.

[11] Z. Pan, J. Lei, Y. Zhang, X. Sun, and S. Kwong, "Fast motion estimation based on content property for low-complexity h. 265/hevc encoder," *IEEE Transactions on Broadcasting*, vol. 62, no. 3, pp. 675–684, 2016.